



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **b tree** and **mapping table**

Found 1,519 of 121,350

Sort results by

Display results

☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1. [Database principles: A mapping mechanism to support bitmap index and other auxiliary structures on tables stored as primary B⁺-trees](#)

Eugene Inseok Chong, Chuck Freiwald, Anh-Tuan Tran, Jagannathan Srinivasan, Aravind Yalamanchi, Ramkumar Krishnan, Souripriya Das, Mahesh Jagannath, Richard Jiang
June 2003 **ACM SIGMOD Record**, Volume 32 Issue 2

 Full text available: [pdf\(198.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Any auxiliary structure, such as a bitmap or a B⁺-tree index, that refers to rows of a table stored as a primary B⁺-tree (e.g., *tables with clustered index* in Microsoft SQL Server, or *index-organized tables* in Oracle) by their physical addresses would require updates due to inherent volatility of those addresses. To address this problem, we propose a mapping mechanism that 1) introduces a single mapping table, with each row holding one key value from the prima ...

2. [Functional-join processing](#)

R. Braumandl, J. Claussen, A. Kemper, D. Kossmann
February 2000 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 8 Issue 3-4

 Full text available: [pdf\(486.22 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Inter-object references are one of the key concepts of object-relational and object-oriented database systems. In this work, we investigate alternative techniques to implement inter-object references and make the best use of them in query processing, i.e., in evaluating functional joins. We will give a comprehensive overview and performance evaluation of all known techniques for simple (single-valued) as well as multi-valued functional joins. Furthermore, we will describe special *order-preser* ...

Keywords: *Functional join, Logical OID, Object identifier, Order-preserving join, Physical OID, Pointer join, Query processing*

3. [Automatic generation of cells for recurrence structures](#)

Avinoam Bilgory, Daniel D. Gajski
June 1981 **Proceedings of the eighteenth design automation conference on Design automation**

 Full text available: [pdf\(478.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a method for automatic translation of functional into structural descriptions for Boolean recurrence systems. The solution of a recurrence system is

accomplished by a network that requires at most four different types of cells. Given any Boolean recurrence of any order, the cell generator module defines the Boolean equations of these cells.

Keywords: Boolean-recurrence solvers, Gate compilers, Logic-design automation

4 Industry session 3: database performance and interface: A mapping mechanism to support bitmap index and other auxiliary structures on tables stored as primary B⁺-trees

Eugene Inseok Chong, Jagannathan Srinivasan, Souripriya Das, Chuck Freiwald, Aravind Yalamanchi, Mahesh Jagannath, Anh-Tuan Tran, Ramkumar Krishnan, Richard Jiang
November 2002 **Proceedings of the eleventh international conference on Information and knowledge management**


Full text available:  [pdf\(63.19 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Any auxiliary structure, such as a bitmap or a B⁺-tree index, that refers to rows of a table stored as a primary B⁺-tree (e.g., *tables with clustered index* in Microsoft SQL Server, or *index-organized tables* in Oracle) by their physical addresses would require updates due to inherent volatility of those addresses. To address this problem, we propose a mapping mechanism that 1) introduces a single *mapping table*, with each row holding one key value from the ...

Keywords: bitmap indexes, mapping mechanism, primary B⁺-trees

5 File organizations and access methods for CLV disks


S. Christodoulakis, D. A. Ford
May 1989 **ACM SIGIR Forum , Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval**, Volume 23 Issue 1-2

Full text available:  [pdf\(1.08 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A large and important class of optical disc technology are CLV format discs such as CD ROM and WORM. In this paper, we examine the issues related to the implementation and performance of several different file organizations on CLV format optical discs such as CD ROM and WORM. The organizations examined are based on hashing and trees. The CLV recording scheme is shown to be a good environment for efficiently implementing hashing. Single seek access and storage utilization levels are ...

6 Optimizing multidimensional index trees for main memory access

Kihong Kim, Sang K. Cha, Keunjoo Kwon
May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2

Full text available:  [pdf\(243.75 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Recent studies have shown that cache-conscious indexes such as the CSB⁺-tree outperform conventional main memory indexes such as the T-tree. The key idea of these cache-conscious indexes is to eliminate most of child pointers from a node to increase the fanout of the tree. When the node size is chosen in the order of the cache block size, this pointer elimination effectively reduces the tree height, and thus improves the cache behavior of the index. However, the pointer elimination cannot be ...

7 Query evaluation techniques for large databases

Goetz Graefe

Full text available:  [pdf\(9.37 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

8 Research sessions: path indexing: Accelerating XPath location steps

Torsten Grust

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**


Full text available:  [pdf\(1.12 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#)

This work is a proposal for a database index structure that has been specifically designed to support the evaluation of XPath queries. As such, the index is capable to support *all* XPath axes (including ancestor, following, preceding-sibling, descendant-or-self, etc.). This feature lets the index stand out among related work on XML indexing structures which had a focus on regular path expressions (which correspond to the XPath axes children and descendant-or-self plus name tests). I ...

9 Special system-oriented section: the best of SIGMOD '94: QuickStore: a high performance mapped object store

Seth J. White, David J. DeWitt

October 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 4

Full text available:  [pdf\(2.58 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#)


QuickStore is a memory-mapped storage system for persistent C++, built on top of the EXODUS Storage Manager. QuickStore provides fast access to in-memory objects by allowing application programs to access objects via normal virtual memory pointers. This article presents the results of a detailed performance study using the OO7 benchmark. The study compares the performance of QuickStore with the latest implementation of the E programming language. The QuickStore and E systems exemplify the two ba ...

Keywords: benchmark, client-server, memory-mapped, object-oriented, performance, pointer swizzling

10 Design of the Mnome persistent object store

J. Eliot B. Moss

April 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 2

Full text available:  [pdf\(3.22 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Mnome project is an investigation of techniques for integrating programming language and database features to provide better support for cooperative, information-intensive tasks such as computer-aided software engineering. The project strategy is to implement efficient, distributed, persistent programming languages. We report here on the Mnome persistent

object store, a fundamental component of the project, discussing design and initial prototype. Mne stores objects

11 Performance comparison of property map and bitmap indexing

Ashima Gupta, Karen C. Davis, Jennifer Grommon-Litton

November 2002 **Proceedings of the fifth ACM international workshop on Data Warehousing and OLAP**

Full text available:  [pdf\(250.60 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A data warehouse is a collection of data from different sources that supports analytical querying. A Bitmap Index (BI) allows fast access to individual attribute values that are needed to answer a query by representing the values of an attribute for all tuples separately, as bit strings. A Property Map (PMap) is a multidimensional indexing technique that pre-computes attribute expressions, called properties, for each tuple and stores the results as bit strings [DD97, LD02]. This paper compares t ...

Keywords: bitmap index, data warehouse, performance study

12 Scalable high-speed prefix matching

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available:  [pdf\(933.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Finding the longest matching prefix from a database of keywords is an old problem with a number of applications, ranging from dictionary searches to advanced memory management to computational geometry. But perhaps today's most frequent best matching prefix lookups occur in the Internet, when forwarding packets from router to router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing user population is increasing the size of routing tables against which p ...

Keywords: collision resolution, forwarding lookups, high-speed networking

13 Data structures for efficient broker implementation

Anthony Tomasic, Luis Gravano, Calvin Lue, Peter Schwarz, Laura Haas

July 1997 **ACM Transactions on Information Systems (TOIS)**, Volume 15 Issue 3

Full text available:  [pdf\(316.45 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

With the profusion of text databases on the Internet, it is becoming increasingly hard to find the most useful databases for a given query. To attack this problem, several existing and proposed systems employ brokers to direct user queries, using a local database of summary information about the available databases. This summary information must effectively distinguish relevant databases and must be compact while allowing efficient access. We offer evidence that one broker, GLOSS

Keywords: GLOSS, broker architecture, broker performance, distributed information, grid files, partitioned hashing

14 Building a scaleable geo-spatial DBMS: technology, implementation, and evaluation

Jignesh Patel, JieBing Yu, Navin Kabra, Kristin Tufte, Biswadeep Nag, Josef Burger, Nancy Hall, Karthikeyan Ramasamy, Roger Lueder, Curt Ellmann, Jim Kupsch, Shelly Guo, Johan Larson, David De Witt, Jeffrey Naughton

June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data**, Volume 26 Issue 2

Full text available:  [pdf\(1.58 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

This paper presents a number of new techniques for parallelizing geo-spatial database systems and discusses their implementation in the Paradise object-relational database system. The effectiveness of these techniques is demonstrated using a variety of complex geo-spatial queries over a 120 GB global geo-spatial data set.

15 Goal-oriented buffer management revisited

Kurt P. Brown, Michael J. Carey, Miron Livny

June 1996 **ACM SIGMOD Record , Proceedings of the 1996 ACM SIGMOD international conference on Management of data**, Volume 25 Issue 2

Full text available:  [pdf\(1.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper we revisit the problem of achieving multi-class workload response time goals by automatically adjusting the buffer memory allocations of each workload class. We discuss the virtues and limitations of previous work with respect to a set of criteria we lay out for judging the success of any goal-oriented resource allocation algorithm. We then introduce the concept of *hit rate concavity* and develop a new goal-oriented buffer allocation algorithm, called *Class Fencing*, th ...

16 Extensions to Starburst: objects, types, functions, and rules

Guy M. Lohman, Bruce Lindsay, Hamid Pirahesh, K. Bernhard Schiefer

October 1991 **Communications of the ACM**, Volume 34 Issue 10

Full text available:  [pdf\(5.21 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: Extended relational database management systems, Starburst, extensible database management systems

17 B-trees: bearing fruits of all kinds

Beng Chin Ooi, Kian-Lee Tan

January 2002 **Australian Computer Science Communications , Proceedings of the thirteenth Australasian conference on Database technologies - Volume 5**, Volume 24 Issue 2

Full text available:  [pdf\(872.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Index structures are often used to support search operations in large databases. Many advanced database application domains such as spatial databases, multimedia databases, temporal databases, and object-oriented databases, call for index structures that are specially designed and tailored for the domains. Interestingly, in each of these domains, we find methods that are based on one distinct structure --- the B-tree. Invented some thirty years ago, the B-tree has been challenged repeatedly, but ...

Keywords: b-tree, high-dimensional databases, main memory databases, multimedia databases, spatial databases

18 An asymptotically optimal multiversion B-tree

Bruno Becker, Stephan Gschwind, Thomas Ohler, Bernhard Seeger, Peter Widmayer

December 1996 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 5 Issue 4

Full text available:  [pdf\(151.97 KB\)](#) Additional Information: [full citation](#), [abstract](#)

In a variety of applications, we need to keep track of the development of a data set over time. For maintaining and querying these multiversion data efficiently, external storage structures are an absolute necessity. We propose a multiversion B-tree that supports

insertions and deletions of data items at the current version and range queries and exact match queries for any version, current or past. Our multiversion B-tree is asymptotically optimal in the sense that the time and space bounds are ...

Keywords: Access methods, Information systems, Physical design, Versioned data

19 Utilization of B-trees with inserts, deletes and modifies

T. Johnson, D. Shasha

March 1989 **Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Full text available:  pdf(1.07 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The utilization of B-tree nodes determines the number of levels in the B-tree and hence its performance. Until now, the only analytical aid to the determination of a B-tree's utilization has been the analysis by Yao and related work. Yao showed that the utilization of B-tree nodes under pure inserts was 69%. We derive analytically and verify by simulation the utilization of B-tree nodes constructed from N inserts followed by M modifies (where M

20 B-trees: Compact B-trees

Arnold L. Rosenberg, Lawrence Snyder

May 1979 **Proceedings of the 1979 ACM SIGMOD international conference on Management of data**

Full text available:  pdf(928.31 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

A B-tree is *compact* if it is minimal in number of nodes, hence has optimal space utilization, among equally capacious B-trees of the same order. The space utilization of compact B-trees is analyzed and is compared with that of noncompact B-trees and of (node)-visit-optimal B-trees, which minimize the expected number of nodes visited per key access. Compact B-trees can be as much as a *factor* of 2.5 more space-efficient than visit-optimal B-trees; and the node-visit cost of a compact ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Searching for **PHRASE b tree**.

Restrict to: [Header](#) [Title](#) Order by: [Citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Amazon](#) [B&N](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

1182 documents found. Only retrieving 125 documents (System busy - maximum reduced). Retrieving documents... Order: citations weighted by year.

Pond: the OceanStore Prototype - Rhea, Eaton, Geels, Weatherspoon.. (2003) (Correct) (3 citations)
by an active GUID, or AGUID. Each version is a B-tree> of read-only blocks child pointers are secure object is stored in a data structure similar to a B-tree>, in which a block references each child by a
oceanstore.cs.berkeley.edu/publications/papers/compressed/fast2003-pond.ps.gz

Information Fusion in Biometrics - Ross, Jain, Qian (2001) (Correct) (3 citations)
4 11,021 Impostor 72 10,953 (a) C5.0 Decision Tree. b) Linear Discriminant classifier. Table 1.
the performance of the (a) C5.0 Decision Tree, and (b) Linear Discriminant classifier, on an
biometrics.cse.msu.edu/RossFusion_AVBPA01.pdf

The Alternating Decision Tree Learning Algorithm - Freund, Mason (1999) (Correct) (8 citations)
semantics for representing b) c) Figure 1: Tree-based classifiers: c) a general alternating tree.
(c) a general alternating tree. a) a decision tree, b) the same decision tree represented as an
is the prediction associated with the root of the tree. b) A base rule r with precondition d can be in the
www.lsmason.com/papers/ICML99-AlternatingTrees.pdf

Multiway Range Trees: Scalable IP Lookup with Fast Updates - Suri, Varghese, Warkhede (2001) (Correct) (2 citations)
be generalized to a multiway search tree we use a B-tree>. In a B-tree>, each node other than the root has
to a multiway search tree we use a B-tree>. In a B-tree>, each node other than the root has at least t
www.cs.ucsd.edu/~varghese/PAPERS/globecom2001.pdf

Implementing I/O-Efficient Data Structures Using TPIE - Arge, Procopiuc, Vitter (2002) (Correct) (1 citation)
been implemented in the second phase, including B-trees [12]persistent B-trees [9]R-trees [10]
second phase, including B-trees [12]persistent B-trees [9]R-trees [10]CRB-trees [1]K-D-B-trees
a data structure using TPIE. We chose the K-D-B-tree> because it is a relatively simple yet typical
www.cs.duke.edu/%7EElarge/Papers/tpieds.ps

Scalable Integrated Region-based Image Retrieval using IRM and.. - Wang, Du (2001) (Correct) (1 citation)
using various tree structures such as K-D-B-tree> [28]quadtree [9] R-tree [11]tree [31]
V, pp. 161-64, 1993. 28] J. Robinson, The k-d-b-tree>: A search structure for large multidimensional
www-db.stanford.edu/~wangz/project/imsearch/SIMPLIcity/DL2001/wang2.pdf

Purely Functional Representations of Catenable Sorted Lists - Kaplan, Tarjan (1996) (Correct) (7 citations)
is an ordinary balanced search tree (like an a,b-tree> for example) in which each node along the left
described could be modified to use any kind of a,b-trees. Particularly interesting is a relation between
returns a tree of the same height as its input tree but it may return a tree in which the root has
www.math.tau.ac.il/~haimk/papers/loglog23.ps

Incremental Processing of Vague Queries in Interactive.. - Pfeifer, Pennekamp (1994) (Correct) (3 citations)
for example be processed very efficiently using a B*tree> for the date attribute. For the string
by the graph in figure 1. The DATE node uses a B*tree>, the AUTHOR node interfaces to a special string
ls6-www.informatik.uni-dortmund.de/bib/fulltext/ir/.Pfeifer_Pennekamp:97.pdf

Probabilistic Top-Down Parsing and Language Modeling - Roark (Correct) (3 citations)
Figure 1 Three parse trees: a) a complete parse tree b) a complete parse tree with an explicit stop
Geman (1998) proved that any PCFG estimated from a treebank with the relative frequency estimator is
First, it is easily reversible, i.e. every parse tree built with Gf corresponds to a unique parse tree

acl.lidc.upenn.edu/J/J01/J01-200

Bitmap Indices for Speeding Up - High-Dimensional Data Analysis (Correct)

optimised for one-dimensional queries such as the Btree whereas others are optimised for S. Berchtold, C. Boehm, H.P. Kriegel, The Pyramid-Tree: Breaking the Curse of D imensionality, SIGMOD kurts.home.cern.ch/kurts/PHD/./RESEARCH/dexa2002_bitmaps.ps

The Rectilinear Steiner Arborescence Problem is NP-Complete - Chen (Correct)

Arborescence (a) a Rectilinear Steiner Minimum Tree (b) and a Rectilinear Minimum Spanning Tree (c) ece.tamu.edu/~wshi/pub/soda.pdf

Slicing Floorplan with Clustering Constraint - Yuen And Evangeline (Correct)

[5] boundsliceline -grid(BSG) 6] O-tree [2] B tree> [1] and TCG [4] have been proposed for Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu. B*trees: A new representation for non-slicing www.cse.cuhk.edu.hk/~fyyoung/paper/tcad5.ps

Versioned Backups and Index Concurrency - Results Of Work-In-Progress (Correct)

of a full temporal index, the Time-Split B-tree> [LoSa1, LoSa2] or TSB-tree with that of a fuzzy new method will undoubtedly be in its use in B trees, it can be extended to other structures. In nodes are split bottom up like those of a B tree>, but can be split either by time or by key. When ftp.research.microsoft.com/users/lomet/pub/newhpts.ps

Dynamically Partitioned Test Scheduling for SoCs Under Power.. - Zhao, Upadhyaya (Correct)

partitioning a test session T2 T1 Time Power (b) tree> growing technique a test session T1 T2 T3 Time www.cse.buffalo.edu/~shambhu/resume/natw02.pdf

Spatial Lesion Indexing for Medical Image Databases Using.. - Histograms Chi-Ren Shyu (Correct)

for objects in two to three dimensional spaces: B-trees [5] R-trees [8] and their variants [2, 20, 21, gray-scale, etc. These tree structures include K-D-B tree> [18] Metric tree [25] and Multihash indexing www.cecs.missouri.edu/~matsakis/Publications/CVPR01.pdf

The BUB-Tree - Robert Fenk Fenk (Correct)

separators of Z-regions in the index part of the B-Tree> it stores Z-intervals bounding the data stored on data, which inherits all good properties of the B-Tree> [BM72] Logarithmic performance guarantees are problem of the UB-Tree, we propose the bounding UB-Tree (BUB-Tree) an UBTree storing additional mistral.in.tum.de/results/publications/Fen02b.pdf

Mixed-Integer Nonlinear Programming - Armin (Correct)

or RMINLP) Moreover, each node of the emerging B&B tree> represents a solution of the RMINLP with adjusted www.gamsworld.org/minlp/siagopt.pdf

A Note on Parallel Algorithms for Optimal h-v Drawings of - Binary Trees Panagiotis (Correct)

i shows three different h-v drawings of the same tree. b e c g c) a) b) d) Figure 1: A binary tree www.wellesley.edu/CS/pmetexas/CGTA98.pdf

Compressing Bitmap Indexes for Faster Search - Operations Kesheng Wu (Correct)

schemes [6] [21] [30] such as the variants of B-tree> [9] or R-tree [11] According to the performance for most database systems. For example, if a B-tree> index is created for each attribute, ORACLE applications, bitmap indexes are better than the tree based schemes [6] [21] [30] such as the variants crd.lbl.gov/~kewu/ps/LBNL-49627.pdf

European Organization For Nuclear Research Cern/lhcc 97-9.. - April Using An (Correct)

18.1 B-Tree>

of indices on selective analysis attributes. 18.1 B-Tree> Indices Indices are a way to cluster one or more number of objects in the tree, this means that a tree based search will scale much better than a wwwinfo.cern.ch/asd/rd45/reports/m3_96/milestone_3.ps

First 20 documents [Next 20](#)

Try your query at: [Amazon](#) [Barnes & Noble](#) [Google \(RI\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide

 SEARCH
THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
B+ Tree Indexes with Hybrid Row Identifiers in Oracle 8i
Source [Proceedings of the 17th International Conference on Data Engineering](#) [table of contents](#)

Pages: 341 - 348

Year of Publication: 2001

ISBN:0-7695-1001-9

Authors [Eugene Inseok Chong](#)
[Souripriya Das](#)
[Chuck Freiwald](#)
[Jagannathan Srinivasan](#)
[Aravind Yalamanchi](#)
[Mahesh Jagannath](#)
[Anh-Tuan Tran](#)
[Ramkumar Krishnan](#)
Publisher IEEE Computer Society Washington, DC, USA

Additional Information: [citations](#) [collaborative colleagues](#)
Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)
↑ CITINGS

[Eugene Inseok Chong](#) , [Chuck Freiwald](#) , [Anh-Tuan Tran](#) , [Jagannathan Srinivasan](#) , [Aravind Yalamanchi](#) , [Ramkumar Krishnan](#) , [Souripriya Das](#) , [Mahesh Jagannath](#) , [Richard Jiang](#) , A mapping mechanism to support bitmap index and other auxiliary structures on tables stored as primary B+-trees, ACM SIGMOD Record, v.32 n.2, June 2003

↑ Collaborative Colleagues:

| | | |
|--------------------------------------|--|------------------------------------|
| Eugene Inseok Chong: | Jayanta Banerjee | Aravind Yalamanchi |
| | Souripriya Das | |
| | Samuel DeFazio | |
| | Chuck Freiwald | |
| | Mahesh Jagannath | |
| | Richard Jiang | |
| | Ramkumar Krishnan | |
| | Peter Scheuermann | |
| | Jagannathan Srinivasan | |
| | Anh-Tuan Tran | |
| Souripriya Das: | Jayanta Banerjee | Anh-Tuan Tran |
| | Eugene Inseok Chong | Aravind Yalamanchi |
| | Samuel DeFazio | |
| | Chuck Freiwald | |
| | Mahesh Jagannath | |
| | Richard Jiang | |
| | Ramkumar Krishnan | |
| | Stephen R. Schach | |



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide

 SEARCH
THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Oracle8i Index-Organized Table and Its Application to New Domains
Source [Proceedings of the 26th International Conference on Very Large Data Bases](#) [table of contents](#)

Pages: 285 - 296

Year of Publication: 2000

ISBN:1-55860-715-3

Authors [Jagannathan Srinivasan](#)
[Jayanta Banerjee](#)[Souripriya Das](#)[Chuck Freiwald](#)[Eugene Inseok Chong](#)[Mahesh Jagannath](#)[Aravind Yalamanchi](#)[Ramkumar Krishnan](#)[Anh-Tuan Tran](#)[Samuel DeFazio](#)
Publisher Morgan Kaufmann Publishers Inc. San Francisco, CA, USA

Additional Information: [citations](#) [collaborative colleagues](#)
Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)
↑ CITINGS

[Eugene Inseok Chong](#) , [Chuck Freiwald](#) , [Anh-Tuan Tran](#) , [Jagannathan Srinivasan](#) , [Aravind Yalamanchi](#) , [Ramkumar Krishnan](#) , [Souripriya Das](#) , [Mahesh Jagannath](#) , [Richard Jiang](#) , A mapping mechanism to support bitmap index and other auxiliary structures on tables stored as primary B+-trees, ACM SIGMOD Record, v.32 n.2, June 2003

↑ Collaborative Colleagues:
[Jayanta Banerjee:](#)[Richard I. Baum](#)[Anh-Tuan Tran](#)[Eugene Inseok Chong](#)[Aravind Yalamanchi](#)[Souripriya Das](#)[Samuel DeFazio](#)[Chuck Freiwald](#)[David K. Hsiao](#)[Mahesh Jagannath](#)[Krishnamurthi Kannan](#)[Ramkumar Krishnan](#)[Jagannathan Srinivasan](#)[Eugene Inseok Chong:](#)[Jayanta Banerjee](#)[Aravind Yalamanchi](#)[Souripriya Das](#)[Samuel DeFazio](#)[Chuck Freiwald](#)[Mahesh Jagannath](#)[Richard Jiang](#)


[Advanced Search](#)
[Preferences](#)
[Language Tools](#)
[Search Tips](#)

b+-tree

Google Search

[Web](#) - [Images](#) - [Groups](#) - [Directory](#) - [News](#)
Searched the web for **b+-tree**.Results **1 - 10** of about **9,250**. Search took **0.53** seconds.

B+-tree

NIST. **B+-tree**. (data structure). ... Author: PEB. Implementation. A **B+-tree** package, b+tree_mjr (C) , with patches b+tree, was posted on comp.sources.misc, volume 10. ...
www.nist.gov/dads/HTML/bplustree.html - 3k - [Cached](#) - [Similar pages](#)

B+ Trees

... Informally, a **B+ tree** is an n-ary tree with n variable but large (often >100). A **B+ tree** ... the actual record. Insertion Into a **B+ Tree**: ...
www.seanster.com/BplusTree/BplusTree.html - 5k - [Cached](#) - [Similar pages](#)

B+ Tree Bookmarks

B+ Tree Bookmarks. Public Bookmarks. Enjoy! BplusTree. <http://www.Seanster.com/BplusTree/>
 CISC 235 Discussion Forum CISC 235 - Information ...
www.seanster.com/BplusTree/source/BplusTreeBookmarks.html - 6k - [Cached](#) - [Similar pages](#)

Sorry, but the database does not contain an entry matching your ...

Animation: **B+ Tree** Insertion. ... File information Title, **B+ Tree** Insertion.
 Animation URL, <http://www.animal.ahrgr.de/Anims/en/bPlusTreeInsert.auml>. ...
www.animal.ahrgr.de/en/Animation8.html - 6k - [Cached](#) - [Similar pages](#)

[PDF]Microsoft PowerPoint - B+ Tree and Hash-Based Index.ppt

File Format: PDF/Adobe Acrobat - [View as HTML](#)
 Page 1. 1 **B+ Tree** and Hash-Based Indexes By Ning Wei Dept of Computer Science
 TAMU Sept 4, 2003 Page 2. ... Page 11. 11 Deleting 24* in **B+ tree** Merge. ...
cocasoft.cs.tamu.edu/~lidu/courses/db03f/slides/BTreeHash.pdf - [Similar pages](#)

B+-Tree Indexes with Hybrid Row Identifiers in Oracle8i

17th International Conference on Data Engineering. April 02 - 06, 2001. Heidelberg, Germany, p. 0341 **B+-Tree** Indexes with Hybrid Row Identifiers in Oracle8i. PDF. ...
www.computer.org/proceedings/icde/1001/10010341abs.htm - 12k - [Cached](#) - [Similar pages](#)

CS432 Assignment 2 : B+ Tree

CS432 - Fall 2001 Assignment 3 - **B+-Tree** Deadline: October 23, 23:59pm. ... Goal
 In this assignment, you are asked to implement a **B+-tree** index file. ...
www.cs.cornell.edu/courses/cs432/2001fa/a3%5Cindex.html - 30k - [Cached](#) - [Similar pages](#)

[PPT]Multilevel Security Index B+ Tree

File Format: Microsoft Powerpoint 97 - [View as HTML](#)
 Multilevel Security Index **B+ Tree**. Nayot Poolsappasit. ... Close (all) channels
 from which the message can be leaked. **B+ Tree** Index: Models. Model: ...
www.cs.colostate.edu/~iray/teach/cs681/nayot.ppt - [Similar pages](#)

[PPT]B+-tree and Hashing

File Format: Microsoft Powerpoint 97 - [View as HTML](#)
B+-tree and Hashing. Model of Computation. Data stored on disk(s). ... **B+-tree**.
 Records can be ordered over an attribute,. SSN, Name, etc. ...
cs-www.bu.edu/faculty/gkollios/ada01/LectNotes/lect2.ppt - [Similar pages](#)

Build a B+-tree

... The B + -tree (sometimes written **B+-tree**, B+tree, or just B-tree) is a variant of the original B-tree in which all records are stored in the leaves and all ...

www.cs.duke.edu/~tavi/btree/ - 6k - [Cached](#) - [Similar pages](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

b+-tree

Google Search

[Search within results](#)

Dissatisfied with your search results? [Help us improve.](#)

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google


[Advanced Search](#)
[Preferences](#)
[Language Tools](#)
[Search Tips](#)

[Web](#) · [Images](#) · [Groups](#) · [Directory](#) · [News](#)

Searched the web for **primary b+-tree "mapping table"**. Results **1 - 9** of about **10**. Search took **0.67** seconds.

Comparison shop for - TABLE

www.mysimon.com Top products for the home from your friendly shopping partner

Sponsored Link

A mapping mechanism to support bitmap index and other auxiliary ...

Sponsored Links

... various auxiliary structures on **primary B + -trees** ... efficient, and (despite the **mapping table** overhead) provides ... Tran , Ramkumar Krishnan, **B+ Tree** Indexes with ... [portal.acm.org/ citation.cfm?id=777000&jmp=abstract&dl=ACM&dl=ACM&CFID=11111111&CFTOKE...](http://portal.acm.org/citation.cfm?id=777000&jmp=abstract&dl=ACM&dl=ACM&CFID=11111111&CFTOKE...) -

[Similar pages](#)

[PDF] A Mapping Mechanism to Support Bitmap Index and Other Auxiliary ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... The translation from the **primary** key to the **mapping table** row identifier is efficiently

done by **primary** key lookup on the **primary B + -tree** structure, which ...

www.acm.org/sigmod/record/issues/0306/D3-Industry-echong.pdf - [Similar pages](#)

[PPT] Sabre Lab Reporting

File Format: Microsoft Powerpoint 97 - [View as HTML](#)

... pin counter and the dirty bit of each frame can be stored in the **mapping table**. ... **Primary** index: if the search key contains the **primary** key ... **B+-tree** Node Structure. ...

lambda.uta.edu/cse6331/l1.ppt - [Similar pages](#)

[PDF] Sabre Lab Reporting

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... dirty bit of each frame can be stored in the **mapping table**. ... **Primary** index: if the search key contains the **primary** key ... $1 < K \leq K_i$ Order p of a **B+-tree** is the ...

lambda.uta.edu/cse6331/l1.pdf - [Similar pages](#)

[PDF] Windows NT File System

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... Ausgewählte Betriebssysteme - NT File System 5 Partition Table • 4 partitions defined in MBR • **Primary** or extended partitions • Extended partition ...

icarus.weber.edu/home/bob/cs3100/lectures/win2k_lectures/Win2KNTFileSystem.pdf - [Similar pages](#)

[PDF] Address Translation Strategies in the Texas Persistent Store

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... factors against using a fine-grained approach as the **primary** address translation ... is likely to significantly increase the size of the **mapping table**, making it ...

www.usenix.org/publications/library/proceedings/coots99/full_papers/kakkad/kakkad.pdf - [Similar pages](#)

[doc] These are REVISIONS of the notes I gatherer in preparation for my ...

File Format: Microsoft Word 6 - [View as HTML](#)

... Partition by: key range, relative record number, or RBA (entry sequence files). **B+ tree primary** index, **B* tree** secondary index. Index partition just like table. ...

Tree Mapping

Find Solutions for Your Business
Free Reports, Info. & Registration
www.KnowledgeStorm.com
Interest: [View](#)

Table tree Prices

Compare Prices, Tax, & Shipping
from online stores at NexTag
www.nextag.com
Interest: [View](#)

[See your message here...](#)

research.microsoft.com/users/Gray/B95.doc - [Similar pages](#)

[\(ps\)Multi-Dimensional Bitmap Indices for Optimising Data Access ...](#)

File Format: Adobe PostScript - [View as Text](#)

... rectangle as the **primary** key. ... $O(|T||A|)$ (3.3) In contrast, the worst case for building a **B+-tree** is given by: ... 1. **Mapping Table** a 00 b 01 c 10 d 11. ...

kurts.home.cern.ch/kurts/RESEARCH/diss.ps.gz - [Similar pages](#)

[\(doc\)Chapter 2 – Background](#)

File Format: Microsoft Word 97 - [View as HTML](#)

... that the joining attribute be the **primary** key of ... chapter, we will be examining the **B+-tree** structure, hereafter ... use of bitmap vectors, a **mapping table**, and a ...

www.cs.ndsu.ndak.edu/~peasland/thesis.doc - [Similar pages](#)

In order to show you the most relevant results, we have omitted some entries very similar to the 9 already displayed.

If you like, you can repeat the search with the omitted results included.

primary b+-tree "mapping table"

Google Search

[Search within results](#)

Dissatisfied with your search results? [Help us improve.](#)

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

 Print Format

Your search matched 1 of 974953 documents.

A maximum of 1 results are displayed, 15 to a page, sorted by **Relevance** in **descending** order.

You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

(hybrid row identifiers)

Results:

Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**1 **B⁺-tree indexes with hybrid row identifiers in Oracle8i**
Eugene Inseok Chong; Souripriya Das; Yalamanchi, A.; Jagannath, M.; Freiwal Srinivasan, J.; Anh-Tuan Tran; Krishnan, R.;

Data Engineering, 2001. Proceedings. 17th International Conference on , 2-6 A 2001

Page(s): 341 -348

[\[Abstract\]](#) [\[PDF Full-Text \(640 KB\)\]](#) **IEEE CNF**